

## *OHJ-1400 Olio-ohjelmoinnin peruskurssi*

**Tentissä ei saa käyttää ylimääräistä kirjallista materiaalia, laskimia, tietokoneita tai muita luntausvälineitä.**

Muutama sana tenttivastauksen kirjoittamisesta:

1. Vastauksessa olet vastaavasi sellaisen ihmisen kysymykseen, joka tuntee kohtalaisen hyvin ohjelmistotekniikan aihealuetta muutoin paitsi tämän kysymyksen osalta. Muista että vastauksesi tarkoitus on vakuuttaa tarkastaja osaamisestasi.
2. Mieti etukäteen esim. ranskalaisilla viivoilla vastauksesi pääkohdat ja lajittele ne johdonmukaiseen järjestykseen — älä kirjoita yhteen pötköön kaikkea mieleen tulevaa.
3. Jos vastaus vaatii ohjelmakoodin kirjoittamista, sen **ei** tarvitse olla pilkulleen syntaksiltaan oikein.

..... Tehtävät 1. & 2. **omalle paperilleen!** Nimi paperiin! .....

1. Alla on selitetty joitain C++:n ja olio-ohjelmoinnin termejä. Mistä termeistä on kysymys? Jos lisäksi selityksessä on jotain viialla, korjaa selitys.

- a) UML:n kaavio, josta käy ilmi ohjelman ajoaikainen käyttäytyminen kokonaisuudessaan.
- b) Jäsenfunktiot (ja -muuttujat), jotka näkyvät vain luokasta periytyneille luokille.
- c) Ne asiat ja toimenpiteet, joita tietyn luokan olioiden on tarkoitus mallintaa ja toteuttaa ohjelmistossa.
- d) Luokan rakentaja, joka ei saa parametreja ja joka luodaan jokaiselle C++:n luokalle automaattisesti.
- e) Ohjelman rakenteen kuvaaminen helpommin tajuttavassa muodossa keskittymällä tiettyihin olennaisiin piirteisiin ja jättämällä kuvauksesta pois vähämerkityksisiä asioita.
- ☞ f) C++:n ilmaisu, jonka jälkeen annettuun luokkaan voi tehdä osoittimia, mutta ☞ luokan muu käyttäminen ei onnistu.

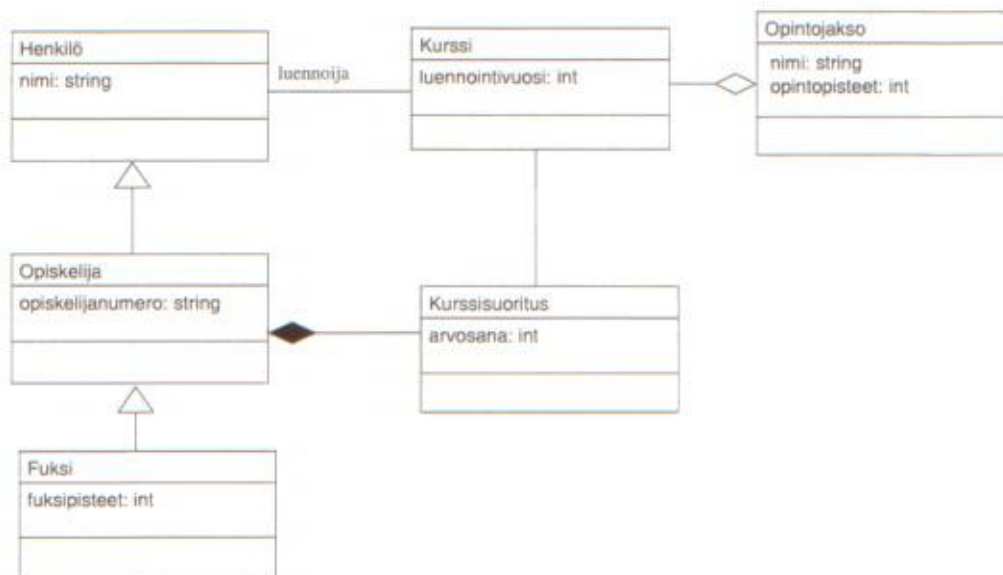
2. Esseitä periytymisestä

- a) Mitä on *periytyminen*? Millaisia mahdollisuuksia yleiskäyttöiseen koodiin periytyminen antaa olio-ohjelmoinnissa?
- b) Mitä tarkoitetaan *dynaamisella sitomisella* ja mitä siihen liittyy?
- c) Miten periytyminen vaikuttaa *rakentajiin ja purkajiin*?

..... **KÄÄNNÄ!** .....

..... Tehtävät 3. & 4. omalle paperilleen! Nimi paperiin! .....

3. Selitä (max. 7 riviä/kohta) seuraavat olio-ohjelmoinnin käsitteet ja *mitä hyötyä/haittaa* niistä on olio-ohjelmoinnissa.
- CRC-kortti (*CRC card*)
  - Nimiavaruus (*namespace*)
  - Koostumissuhde (*aggregation*)
  - Kopiorakentaja (*copy constructor*)
  - Luokkahierarkia (*class hierarchy*)
  - Abstrakti kantaluokka (*abstract base class*)
4. Alla on osa TTKK:n kursseja ja niiden suorituksia kuvaavaa luokkakaaviota (rajapinnat on yksinkertaisuuden vuoksi jätetty pois). Vastaa sen perusteella seuraaviin kysymyksiin:
- Mitä tarkoittaa Opintojakson ja Kurssin välinen "avoin salmiakki"?
  - Mahdollistaako kaavio sen, että jonkin kurssin opiskelija voisi toimia myös luennoijana? Perustele vastauksesi.
  - Kopioi kaavio vastauspaperille ja lisää luokkien väliin suhteisiin lukumäärämerkinnät ja tarvittaessa suunnat. Perustele lyhyesti valintasi.
  - Kaavion alla on hahmotelma kaavion Kurssi-luokasta C++:lla. Kirjoita private-puolelle kaavion perusteella järkevät jäsenmuuttujien esittelyt. Perustele valintasi.
  - Lisää puuttuvat const-sanat luokan esittelyyn (riittää, että kopioit vastauspaperille rivinumeroineen ne rivit, joihin tulee muutoksia).
  - Onko luokan esittelyssä virheitä tai kyseenalaisuuksia? Korjaa ne.



```

1 #include <string>
2 #include <iostream>
3 using namespace std;
4
5 class Kurssi : public Opintojakso
6 {
7     Kurssi();
8     int milloin_luennoitu();
9     string kuka_luennoi();
10
11     void vaihda_luennoija(Henkilo uusi_luennoija);
12     string anna_kurssin_nimi();
13     int anna_opintopisteet();
14     void lisaa_suuritus(Kurssisuoritus* suoritus);
15     // ...
16     private:
17     // ?
18 };
  
```